

SIMULINK MODEL CODE GENERATION FOR MOTOR CONTROL APPLICATIONS

Lukáš Otava

Doctoral Degree Programme (2), FEEC BUT

E-mail: xotava01@stud.feec.vutbr.cz

Supervised by: Pavel Václavek

E-mail: vaclavek@feec.vutbr.cz

Abstract: This article is focused on the embedded software development using code generation. C-code is generated from Simulink model. Article describes hardware interface of Simulink model for AC motor control applications code generation. Results are presented on platform with ARM Cortex-M4 micro-controller, inverter and permanent magnet synchronous machine. Measurements of speed control loop on a real machine are presented and utilization of used micro-controller are discussed in conclusion.

Keywords: Motor Control, Code generation, Simulink Coder, ARM Cortex-M4, STM32F3, PMSM, Vector control

1 INTRODUCTION

Today's AC machines are controlled by micro-controllers and digital signal processors or by programmable logic devices. This article is focused on software development by automatic C-code generation from Simulink model for micro-controllers with signal processing capabilities.

1.1 CODE GENERATION ADVANTAGES AND DISADVANTAGES

Development of advanced control algorithms for AC motors is actual task due their frequent application. In traditional V diagram work-flow, embedded systems are usually developed with help of simulations. After this phase, time consuming implementation phase is established. Final phase of work-flow is verification. Verification must be done on a real hardware is final phase. On the other hand, waterfall diagram work-flow is more suitable for complex embedded projects. This work-flow implements and verifies each step of work-flow. AC motor control algorithms are more suitable for this work-flow and code generation process.

Code generation offers advantages in reduction of implementation time and advantages in code compactness. But there are additional disadvantages related to price of specific tools and need for target platform support packages.

1.2 AC MOTOR CONTROL APPLICATIONS SPECIFICS

There are many common factors in motor control which are in some way specific for control loops. These specifics are usually close-knitted to the power electronic which is used in switching-mode voltage converters. [2]

Switching-mode voltage converters use pulse width modulation (PWM) for their effective operation. Frequency of PWM signal is determined by semiconductor's parameters (efficiency point of view) and load of inverter (current ripples, audible noise). Sampling frequency of motor control loops is determined by electrical time constants of a load and it is usually derived from the PWM frequency

in integer ratio. For current control loops, phase current analogue-to-digital conversion (ADC) have to be synchronized with PWM signal. Because PWM signal is usually complementary and phase currents can be measured only in the specific part of duty cycle there are therefore demands for hardware with specific synchronization modes for synchronization between peripherals. Precise current measurement is required for suppressing jitter in signals.

AC machines are usually controlled from rotor point of view. Sinusoidal stator quantities are seen from rotor as constants in steady state. This scheme is called vector control and also require computational power for mathematical transformations. Shaft position and speed measurement is required for traditional vector control of AC motors. This measurement have to be precise and there are complications with speed measurement. This also require special utilization of hardware peripherals and its synchronization. [2]

1.3 TARGET SUPPORT AND DRIVERS FOR CODE GENERATION

If we want to use code generator in embedded world we have to use some kind of interface which provide connection of generated code with hardware layer.

This is done by target support packages for Embedded coder in Simulink. This packages are available for many different platforms [1]. But these packages include only basic modes of peripherals. Synchronization between peripherals is usually omitted. Motor control loops are difficult to implement without peripherals synchronization, therefore there is a demand for custom drivers with peripheral synchronization for motor control. Custom drivers are also more suitable for production hardware from performance point of view.

2 HARDWARE PLATFORM OVERVIEW

Traditional vector control of AC motor requires sensors for measurement (shaft position, velocity, phase currents) and hardware representing voltage source inverter. For motor control development, a simple voltage source inverter printed circuit board with these interfaces was constructed. There are mosfet drivers from IR, mosfets from NXP and shunt resistor phase current measurement.

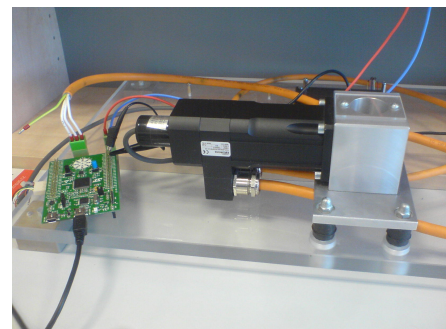
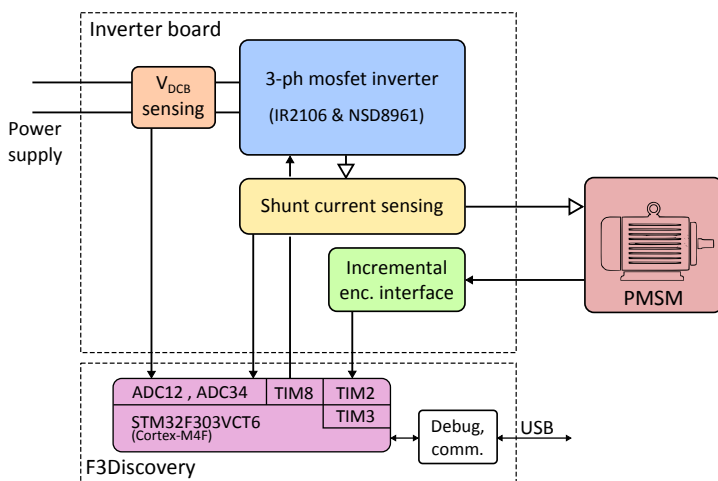


Figure 1: Motor control voltage source inverter

This voltage source inverter is controlled by STM32F3Discovery development board from ST Microelectronics with ARM Cortex-M4 micro-controller and debug interface [4]. This micro-controller

is equipped with single-precision floating point unit [3]. Block diagram and photo of the development platform can be seen in figure 1.

Because every motor control scheme contains similar interfaces, simple C API for motor control applications was created. API contains initialization function `initMotorHW()`, functions for measurement: `motorGetPositionEl()`, `motorGetSpeed()`, `motorGetADCvalues()` and function `motorSetAlfaBeta()`, which sets voltage on PWM inverter output. The API guarantee modular design of motor control drivers, which is required for interfacing with generated code.

3 MATLAB/SIMULINK EMBEDDED CODER TOOL

Simulink Embedded Coder is integrated into *Simulink model editor*. All settings of Embedded coder are in *Model configuration parameter* dialogue window.

Key settings for code generation are in *System target file*. This file determines a process of code generation, parameters of target platform and tools for target platform. For code generation with custom HW drivers, generic system target file `ert.tlc` was used. In *Interface* setting pane of code generation, *Software environment* was set to support only floating point numbers and absolute time. After successful options setting, the code generation process can be started with *Build model* button from tool bar. [1]

3.1 CUSTOM HW DRIVER BLOCKS INTEGRATION

In Simulink models, user can define own blocks using S-functions. But these S-functions aren't directly usable in code generation process. TLC language is a tool for scripting during code generation process. There is a possibility to write TLC file, which describes, how to substitute S-function with C-code for target platform during code generation process. TLC file contains a recipe, which C-code should be placed into definition section, which C-code substitutes initialization function, output function of S-function and so on. All HW drivers can be integrated in this way. Example with CMSIS DSP function integration is outlined in figure 2.

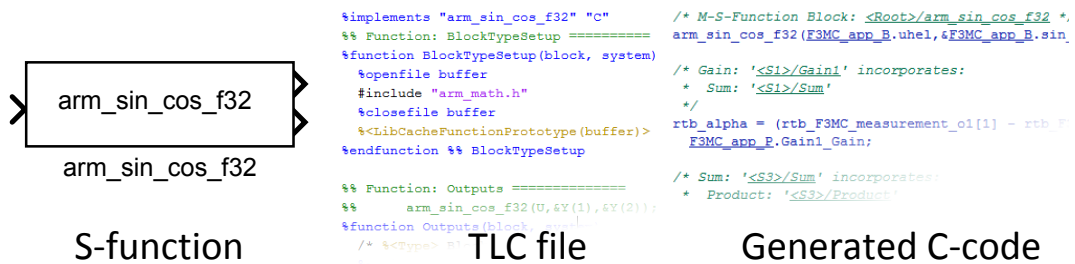


Figure 2: Process of user code integration into generated code

For motor control purposes, functions from Motor Control API and `arm_sin_cos_f32()` function from ARM CMSIS DSP library were interfaced in discussed way. S-functions representing these blocks define signals with single precision. Library of implemented Simulink blocks is shown in figure 3. *F3MC_alfaBetaToSVM* block ensures periodic call of `rt_OneStep()` function. Parameter of this block specifies PWM frequency and therefore the real base sampling frequency of model.

In case of platform specific implementation of mathematical functions, *Code replacement library* setting of Embedded coder interface should be used. This feature can map block inputs and outputs to function call with respect to signal types. Input and output signals could be also accessed through C pointers.

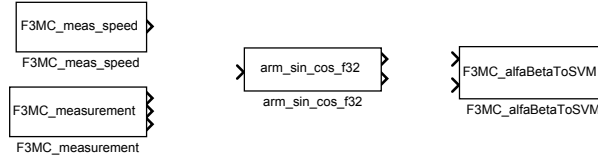


Figure 3: F3MC drivers in Simulink model

4 TESTS AND PERFORMANCE ANALYSIS

HW drivers and whole code generation process were tested with traditional speed vector control scheme for permanent magnet synchronous machine. This vector control scheme contains speed control loop, current control loop and signal transformations [2]. Final model for code generation is depicted in figure 4. All controllers in the model are discrete PI type designed for specific motor parameters. PWM frequency and current control sampling frequency were set to 20 kHz. Speed control loop has sampling frequency 2 kHz.

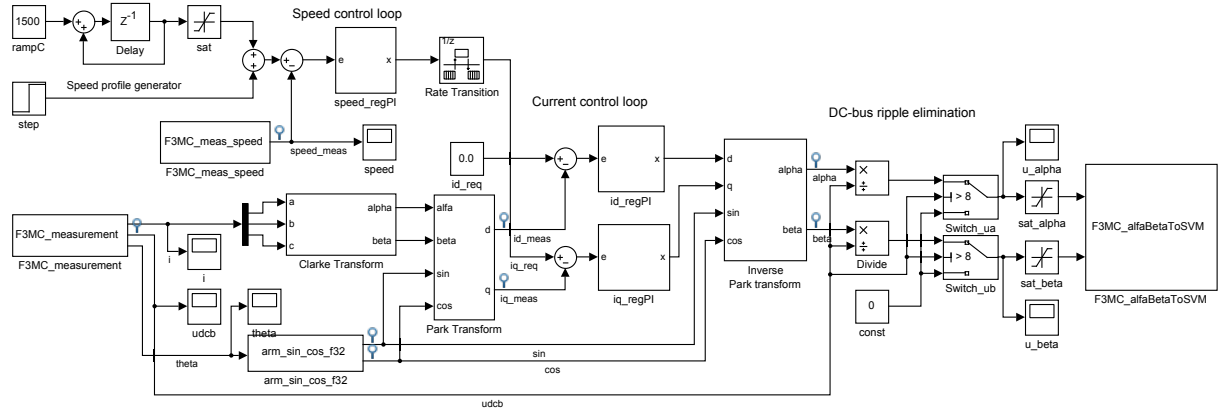


Figure 4: Simulink model for code generation

Matlab version R2014b was used for code generation. Generated code was included in micro-controller project in CoIDE 1.7.7 integrated development environment. This project contain supporting files for micro-controller and also HW driver code and ARM CMSIS DSP library. Compilation of sources was done by gcc-arm-embedded tools version 4.8 with $-Os$ optimization setting.

4.1 MEASURED DATA FROM REAL SYSTEM

Data from model running on target platform was recorded with STM Studio debugging software. STM Studio software communicates with target micro-controller through debugging interface and facilitate access to variables in memory of micro-controller in a direct mode (online) or in a snapshot mode (offline). All variables were measured in *Snapshot mode*. Snapshot mode requires additional SW in micro-controller, which saves recorded variables into structure in micro-controller memory. This is done in each sample period after trigger condition. Measured responses of speed and currents to speed step command 100 rad/s are shown in figure 5.

Performance of generated control algorithm was measured by oscilloscope on toggled GPIOs of micro-controller. Micro-controller was clocked at 72 MHz. Duration of current control loop with transformations is $23.6 \mu s$ of micro-controller time. Duration of speed control loop is $9.2 \mu s$, which is total $32.8 \mu s$ of micro-controller time each 10th current control sampling period.

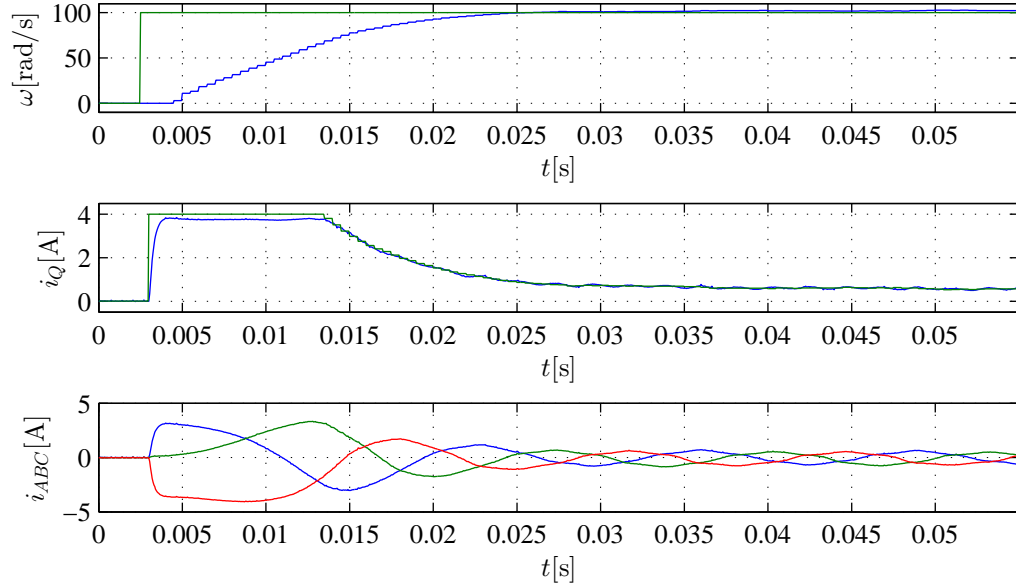


Figure 5: Response to speed step command 100 rad/s

5 CONCLUSION

Process of embedded software development for motor control application with help of code generation tools was presented in this article. Article also discusses development of custom HW interface in situations when target support package is not available or doesn't fit the target or a production platform.

A permanent magnet synchronous machine speed control application was successfully developed using a custom drivers and Simulink Embedded Coder. Measurement on a real target platform reflects working example with micro-controller utilization about 50 %. These results point out, that the presented platform can be practically used for motor control algorithm development and testing.

ACKNOWLEDGEMENT

The completion of this paper was made possible by grant No. FEKT-S-14-2429 - "The research of new control methods, measurement procedures and intelligent instruments in automation", and the related financial assistance was provided from the internal science fund of Brno University of Technology.

REFERENCES

- [1] MATHWORKS, Inc. *Code Generation from Simulink Model: Documentation center*. Version R2014b. 2014.
- [2] ONG, Chee-Mun. *Dynamic Simulations of Electric Machinery: Using MATLAB/SIMULINK*. Prentice Hall, 1997. ISBN 978-0137237852.
- [3] YIU, Joseph. *The Definitive Guide to Arm® Cortex®-M3 and Cortex®-M4 Processors*. B.m.: Elsevier, 2014 [accessed. 15. September 2014]. ISBN 9780124080829.
- [4] STMICROELECTRONICS. *STM32F302xx, STM32F303xx and STM32F313xx advanced ARM-based 32-bit MCUs: Reference manual*. 2013, RM0316.